IRIS: I/O Redirection via Storage Integration

Anthony Kougkas, PhD candidate
Hariharan Devarajan, PhD student
Professor Xian-He Sun

akougkas@hawk.iit.edu
hdevarajan@hawk.iit.edu
sun@iit.edu

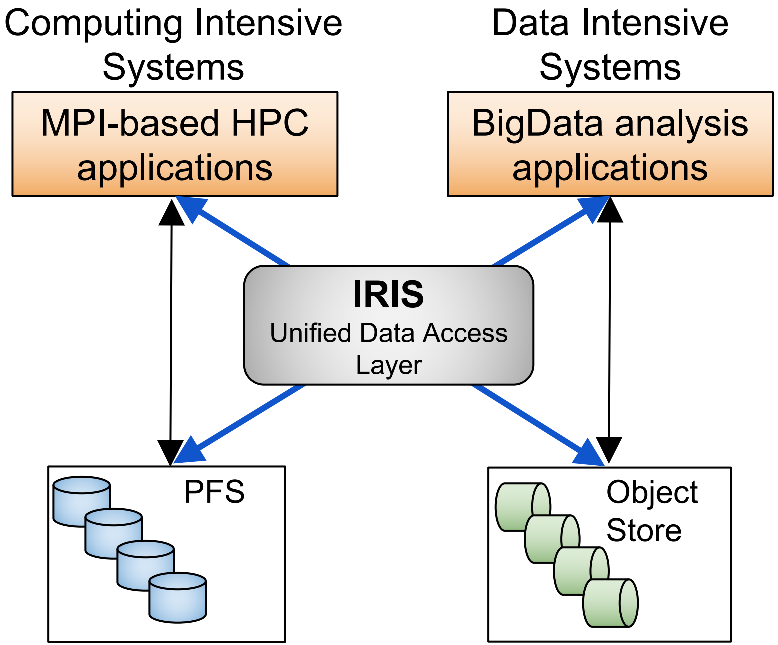
IRIS overview

Objectives:

- Enable MPI-based applications to access and store data in an Object Store.
- Enable HPDA-based applications to access and store data in a PFS.
- Enable a hybrid storage access layer agnostic to files or objects.

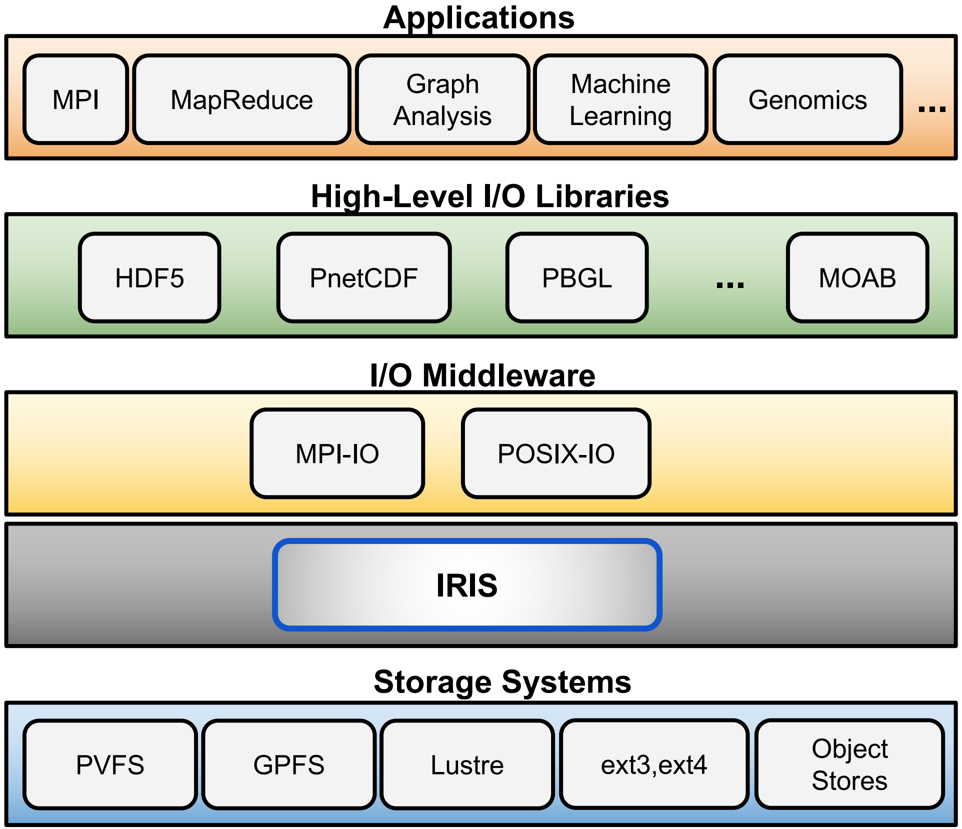
Goal:

- Increase productivity, performance, and resource utilization.


Computing Intensive Systems
Data Intensive Systems
MPI-based HPC applications
BigData analysis applications
IRIS
Unified Data Access Layer
PFS
Object Store

IRIS features

- Middleware library
- Seamless integration to applications (link to IRIS)
- Currently supports:
  - POSIX and MPI-IO
  - HDF5 and pNetCDF
  - S3 and Openstack Swift
  - MongoDB and Hyperdex
- Tunable data consistency
- Relaxed metadata ops
- Caching within IRIS
- Prefetching for faster read
- Non-blocking I/O



The challenge

The tools and cultures of high-performance computing and big data analytics have diverged, to the detriment of both; unification is essential to address a spectrum of major research domains.

- DANIEL A. REED

-JACK DONGARRA

IRIS on Chameleon Testbed

Hardware setup:

- Bare metal configuration
- Each node has:
  - dual Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz (i.e., a total of 48 cores per node)
  - 128 GB RAM
  - 10Gbit Ethernet,
  - local 200GB HDD.
- The total experimental cluster consists of:
  - 1536 client MPI ranks (i.e., 32 nodes),
  - and 16 server nodes
- All test results are the average of five repetitions to eliminate OS noise.

Software used:

- The operating system of the cluster is CentOS 7.0, MPI version is Mpich 3.2,
- PFS is OrangeFS 2.9.6, Object Store is MongoDB 3.4.3.
- **CM1**: a three-dimensional, non-hydrostatic, non-linear, timedependent numerical model designed for idealized studies of atmospheric phenomena,
- **Montage**: an astronomical image mosaic engine,
- **WRF**: a next-generation mesoscale numerical weather prediction system designed for both atmospheric research and operational forecasting needs,

Evaluation Workflow

Total time =

SimulationWrite +

Convert&CopyData fromPFStoKVS +

DataAnalysis +

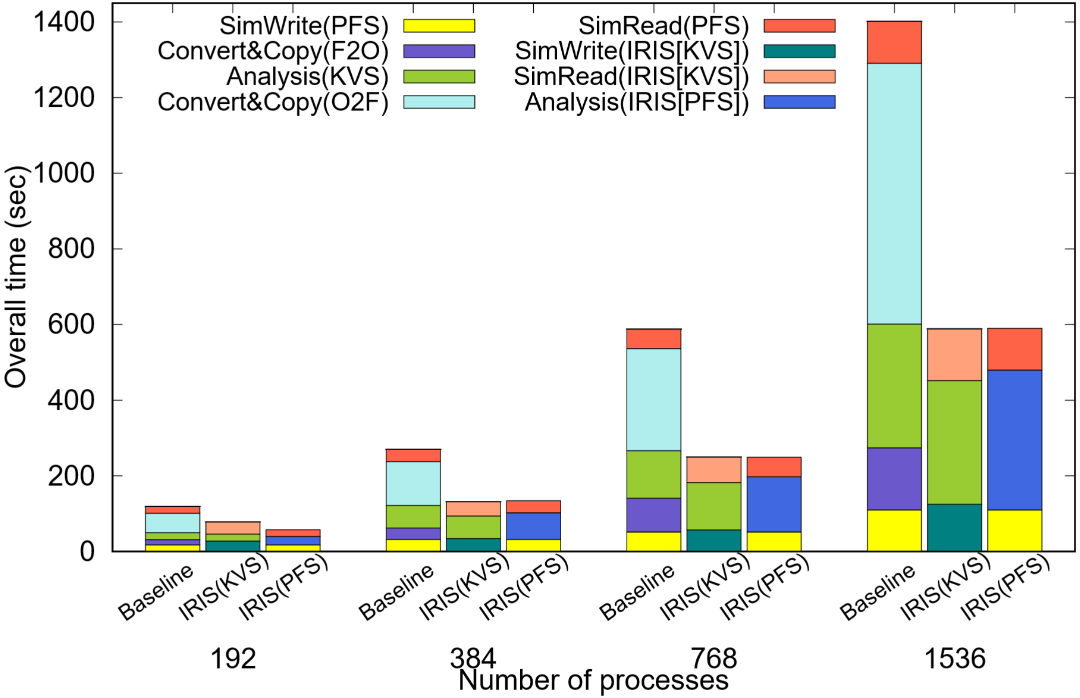
Convert&CopyData fromKVStoPFS +

SimulationRead.

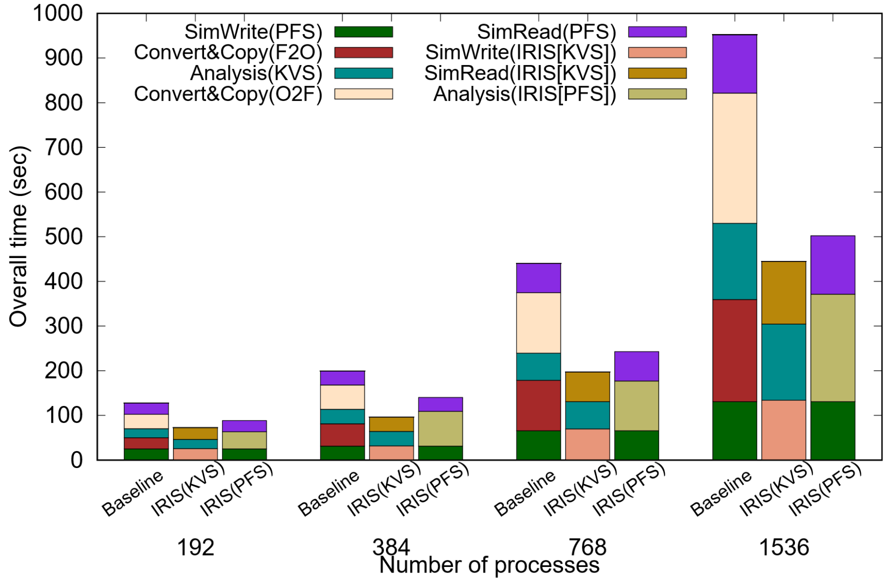
Results

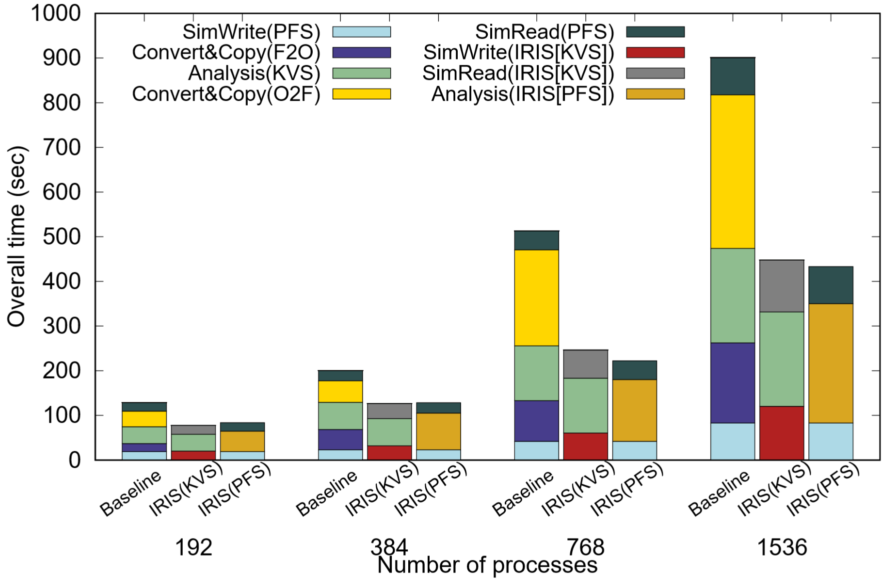
CM1:

- In this test, every process:
  - first writes the checkpoint data
  - then data are combined with observation data residing on the KVS
  - Data are analyzed with a Kmeans clustering kernel.
  - Finally the analysis results are fed back to the simulation as an input for the next phase.
- The performance gain is more than 2x.



Results





Same trend for Montage (left) and WRF (right).

Performance gain is more than 2x.

Challenges

- Scaling of OrangeFS and MongoDB
    - Kept failing after a certain scale (> 1024 client ranks)
    - Performed a lot of optimizations (network, sockets, multithreaded servers)
- Starting instances was often problematic! Be patient ☺
- Some instances were suddenly dying! Solution? Simply restart ☹
- Installing a lot of software on our own OS and then snapshotting it is time consuming: 1 week lease -> first 2 days setting the environment!

○ Maybe have an NFS somewhere?

Conclusions

- By bridging the semantic gap between files and objects, **IRIS** can bring us closer to the convergence of HPC and BigData Analytics.
- Experimental evaluations show that, in addition to providing programming convenience and efficiency, **IRIS** can grant more than 2x higher performance.

Chameleon proved to be a very capable cluster. It requires a learning curve but after this, it is really a pleasure working with the diverse hardware capabilities it offers.
Thumbs up to the team. Every ticket was addressing promptly! ☺